

APELLIDOS, Nombre:

Nº Matrícula:

---

**UPM ETSIInf. Departamento de Lenguajes y Sistemas Informáticos e Ingeniería del Software.**  
*Examen de Programación II. 7 de Mayo de 2015. Examen Parcial.*

---

**Realización:** El test se realizará en la hoja de respuesta. Es **importante** rellenar los datos personales y el código clave de examen del enunciado. Se pueden utilizar hojas aparte como borrador.

**Duración:** La duración total del test será de **50 minutos**.

**Puntuación:** El test se valora sobre **10 puntos**. Las preguntas tipo test pueden tener una única respuesta o varias respuestas, el enunciado lo deja claro. Cada pregunta con una única respuesta respondida correctamente vale 1 punto, e incorrectamente respondida resta 1/3 puntos. Si en una pregunta con una única respuesta se selecciona más de una respuesta, la pregunta se puntuará con 0 puntos. Para una pregunta con varias respuestas, cada afirmación correcta seleccionada suma  $1/n^{\circ}_{respuestas\_correctas}$  puntos, y cada afirmación incorrecta seleccionada resta  $1/n^{\circ}_{respuestas\_incorrectas}$  puntos. Las preguntas no contestadas suman 0 puntos en cualquier caso.

**Calificaciones:** Las calificaciones se publicarán en la plataforma Moodle de la asignatura como muy tarde el día **11 de Mayo de 2015**.

**Revisión:** Las revisiones serán los días **12 y 13 de Mayo de 2015** de 11:00h a 13:00h en el despacho D-2315.

**Prueba eliminatoria:** Este test es eliminatorio y habrá que sacar una **nota mínima de 3 puntos** para poder realizar el examen de tipo problema del día **18 de Mayo de 2015**.

---

**Primer Ejercicio****Las soluciones están marcadas en amarillo**

Al desarrollar una clase, por ejemplo MiClase:

**Pregunta 1**

Indicar cómo se debe declarar el constructor. **Sólo una respuesta es correcta.**

- A)** public MiClase(...) {...}
  - B) public static void MiClase(...) {...}
  - C) public static MiClase(...) {...}
  - D) public void MiClase(...) {...}
- 

**Segundo Ejercicio**

Dado el siguiente fragmento de código, indicar cuál será la salida por consola al ejecutar el *main()*:

```
class Numero {
    private int valor;
    public Numero(int valor) {
        this.valor = valor;
    }
    public Numero(Numero numero) {
        valor = numero.valor;
    }
    public void duplicar() {
        valor = 2*valor;
    }
    public String toString() {
        return "" + valor;
    }
    public static void main(String[] args) {
        Numero a = new Numero(3);
        Numero b = new Numero(a);
        Numero c = b;
        a.duplicar();
        b.duplicar();
        c.duplicar();
        System.out.println(a + ", " + b + ", " + c);
    }
}
```

**Pregunta 2**

Marcar la salida correcta. **Sólo una respuesta es correcta.**

- A)** 6, 12, 12
  - B) 12, 12, 6
  - C) 6, 6, 6
  - D) 12, 12, 12
-

**Tercer Ejercicio**

Sean las siguientes afirmaciones sobre objetos y clases.

**Pregunta 3**

Señalar **todas** las afirmaciones verdaderas. **Puede haber más de una afirmación correcta.**

- A) Los métodos definen el comportamiento de los objetos.**
- B) Los atributos definen el comportamiento de los objetos.
- C) Los atributos definen el estado (contenido de información) de los objetos.**
- D) Los métodos definen el estado (contenido de información) de los objetos.

**Cuarto Ejercicio**

En relación con el uso de paquetes para desarrollar programas modulares en Java:

**Pregunta 4**

Señalar **todas** las afirmaciones verdaderas. **Puede haber más de una afirmación correcta.**

- A) Puede haber dos clases con el mismo nombre en el mismo paquete.
- B) Puede haber dos clases con el mismo nombre en distintos paquetes.**
- C) Puede haber dos métodos con el mismo nombre en una misma clase.**
- D) No puede haber dos métodos con el mismo nombre en clases distintas del mismo paquete.

**Quinto Ejercicio**

Si un objeto posee atributos que son a su vez objetos, hay que tener cuidado para evitar manejar las referencias como si fueran valores. El siguiente código pretende definir una clase Persona con los siguientes requisitos:

1. Los objetos referenciados por los atributos no deben ser modificados una vez construido el objeto Persona. Es decir, no debe ser posible acceder al objeto referenciado por un atributo desde fuera de la clase.
2. Dos personas se consideran iguales si coinciden sus nombres y sus fechas de nacimiento.

```
public class Persona {
    private String nombre;
    private Fecha fechaNacimiento;
    public Persona(String nombre, Fecha fechaNacimiento) {
        this.nombre = nombre;
        this.fechaNacimiento = fechaNacimiento;
    }
    public Fecha getFechaNacimiento() {
        return fechaNacimiento;
    }
    public boolean esIgual(Persona otra) {
        return nombre == otra.nombre && fechaNacimiento.esIgual(otra.fechaNacimiento);
    }
}
```

El código anterior no es totalmente correcto, ya que en algún caso no cumple estrictamente con lo que se ha especificado. Se supone que la implementación de Fecha es totalmente adecuada, y por tanto no es la causa de las posibles deficiencias.

**Pregunta 5**

Indicar cuál de las siguientes afirmaciones es verdadera. **Sólo una respuesta es correcta.**

- A) La implementación de *esIgual()* es correcta y cumple con el requisito 2.
- B) Ninguna de las implementaciones de métodos es totalmente correcta.**
- C) La implementación de *getFechaNacimiento()* es correcta y cumple con el requisito 1.
- D) La implementación del constructor es correcta y cumple con el requisito 1.

**Sexto Ejercicio**

Dado el siguiente código:

```
public class Fecha {
    private int dia, mes, año;
    public Fecha(int d, int m, int a) {
        dia = d; mes = m; año = a;
    }
    ...
}
public class EjercicioArray {
    public static void main(String args[]) {
        Fecha[] miArray = null;
        for (int i = 0; i < miArray.length; i++) {
            miArray[i] = new Fecha(1, 1, 2000+i);
        }
    }
}
```

**Pregunta 6**

Señalar **todas** las afirmaciones verdaderas respecto al método *main()*. **Puede haber más de una afirmación correcta.**

- A) El código compila pero produce un error de ejecución.**
- B) El código compila y se ejecuta sin errores.
- C) El código no compila.
- D) El código compila y se ejecuta sin errores si declaramos `Fecha[] miArray = new Fecha[5];`**

**Séptimo Ejercicio**

Respecto a los programa de prueba:

**Pregunta 7**

Señalar **todas** las afirmaciones verdaderas. **Puede haber más de una afirmación correcta.**

- A) Se debe incluir el código de prueba en la clase probada y no en una clase aparte.
- B) Se debe comprobar que el resultado obtenido es el correcto.**
- C) Se debe hacer una sola prueba que pruebe todo y no varias pruebas individuales.
- D) Se deben probar casos límite, y no solo casos normales.**

**Octavo Ejercicio**

El mecanismo de herencia en Java permite definir una clase hija (subclase) a partir de otra clase padre (superclase).

**Pregunta 8**

Señalar **todas** las afirmaciones verdaderas. **Puede haber más de una afirmación correcta.**

- A) El código de la subclase debe incluir una copia de los métodos de la superclase, de lo contrario se producirá un error de compilación.
- B) El código de la subclase puede usar los atributos *protected* de la superclase.**
- C) El código de la subclase puede usar los métodos públicos de la superclase.**
- D) El código de la subclase puede usar los métodos privados de la superclase.

**Noveno Ejercicio**

En relación con el mecanismo de excepciones en Java.

**Pregunta 9**

Señalar **todas** las afirmaciones verdaderas. **Puede haber más de una afirmación correcta.**

- A) Si se produce una excepción y no se captura el programa continúa ejecutándose con normalidad.
- B) La estructura *try-catch* permite capturar y tratar excepciones.**
- C) En java no existe ninguna sentencia para lanzar una excepción, ya que todas las excepciones son lanzadas de forma automática por la máquina virtual.
- D) Es apropiado usar excepciones para señalar que una operación no puede ejecutarse por no cumplirse la precondition.**

**Décimo Ejercicio**

El siguiente código intenta simplificar la implementación de la cadena enlazada de enteros vista en clase:

```
public class CadenaEnteros {
    private int primero;
    private CadenaEnteros resto;
    public CadenaEnteros(int primero, CadenaEnteros resto) {
        this.primero = primero;
        this.resto = resto;
    }
    public String toString() {
        String imagen = "";
        CadenaEnteros x = this;
        while (x != null) {
            imagen = imagen + "<" + x.primero;
            x = x.resto;
        }
        return imagen;
    }
}
```

Se dispone también del siguiente programa de prueba:

```
public static void main(String[] args) {
    CadenaEnteros cadena = null;
    cadena = new CadenaEnteros(1, cadena);
    cadena = new CadenaEnteros(2, cadena);
    cadena = new CadenaEnteros(3, cadena);
    System.out.println(cadena);
}
```

**Pregunta 10**

Indicar cuál es la salida por la consola al ejecutar el *main()*. Sólo una respuesta es correcta.

- A) < 1 < 2 < 3
- B) El código no compila.
- C) El código compila pero se produce error de ejecución.
- D) <3<2<1**